ECSE 2920: Design Methodology

Deliverable #8

Group #11

03/12/2025

Part 1: Finalized Filter Design

Requirements:

- Implement and test the hardware filter using the +/- 12V power supply
- Verify the hardware filter design with sound or a function generator as input.
- Create an experimental Bode plot using a spectrum analyzer
- Ensure the filter isolates the C8 frequency (4186 Hz) and ignores unwanted signals
- Discuss differences between the theoretical and experimental filter results

Attempt 1: Infinite gain Multiple Feedback Active Filter

Our first approach was to implement an infinite gain multiple feedback active filter, which was selected due to its high gain and ability to target specific frequencies. This filter is commonly used for band-pass applications and was expected to be a suitable choice for isolating the C8 frequency from a microphone input.

We initially found this design online and implemented it in Multisim. The circuit diagram for this filter can be seen in **Figure 1**. We then constructed the circuit in Multisim, as seen in **Figure 2**. After simulating the circuit in Multisim, we generated a theoretical Bode plot, which is shown in **Figure 3**.

From the theoretical results, the filter seemed to perform well in isolating 4186 Hz, so we proceeded to build the physical circuit and test it with a microphone input. The physical setup for testing the circuit can be seen in **Figure 4**. The resulting bode plot can then be seen in **Figure 5**.

Once the circuit was constructed, we attempted to analyze its response to an audio signal. However, we encountered several issues. The experimental Bode plot did not match our expectations, showing erratic frequency response rather than a sharp band-pass effect. While the exact cause of this failure was unclear to us, some potential reasons could have been due to component tolerances, 741 op-amp limitations in handling high-frequency signals effectively, or just unexpected loading effects from the microphone input. Ultimately, due to these inconsistencies, we decided to abandon this design and pursue a different approach.

Attempt 2: Narrow Band-pass Filter

Given the shortcomings of the previous design, we opted for a Narrow Band-pass Filter, which provides sharper frequency isolation and is more suited for our application. The new circuit diagram can be seen in **Figure 6**. To determine the optimal component values, we used an online Narrow Band-pass filter Design Calculator, which provided the values seen in **Figure 7**. The calculated values were chosen to achieve a center frequency of 4186 Hz, with a mid-band gain of 5 and a quality factor of 8.372.

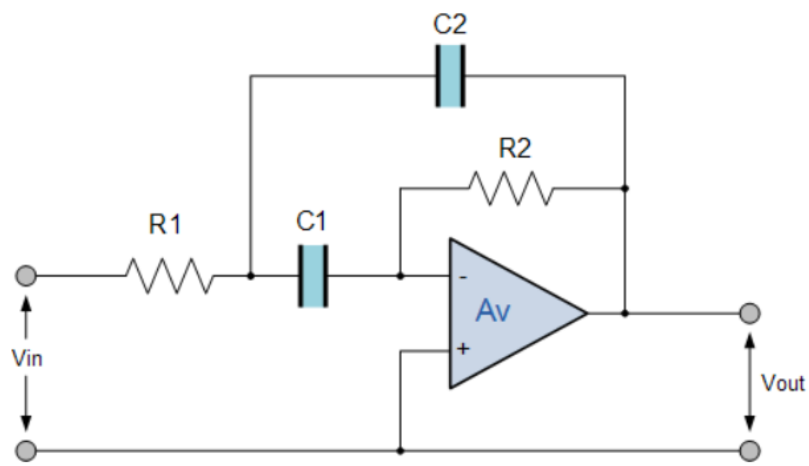
After designing the circuit, we simulated it in Multisim. The Multisim circuit setup can be seen in **Figure 8**. The simulation results then produced the bode plot that can be seen in **Figure 9**. Since the theoretical response matched our expectations, we proceeded to construct the physical circuit.

Before testing with an actual microphone input, we first verified the filter response using a spectrum analyzer with a function generator with a 4186 Hz frequency as input. Our op-amp was also receiving voltage from our push-pull that we constructed in previous deliverables. The setup for the spectrum analyzer testing can be seen in **Figure 10**. The resulting frequency response from the spectrum analyzer can then be seen in **Figure 11**. We applied the 4186 Hz signal and observed that the filter successfully amplified and isolated this frequency while attenuating others, confirming that the design was functioning as expected.

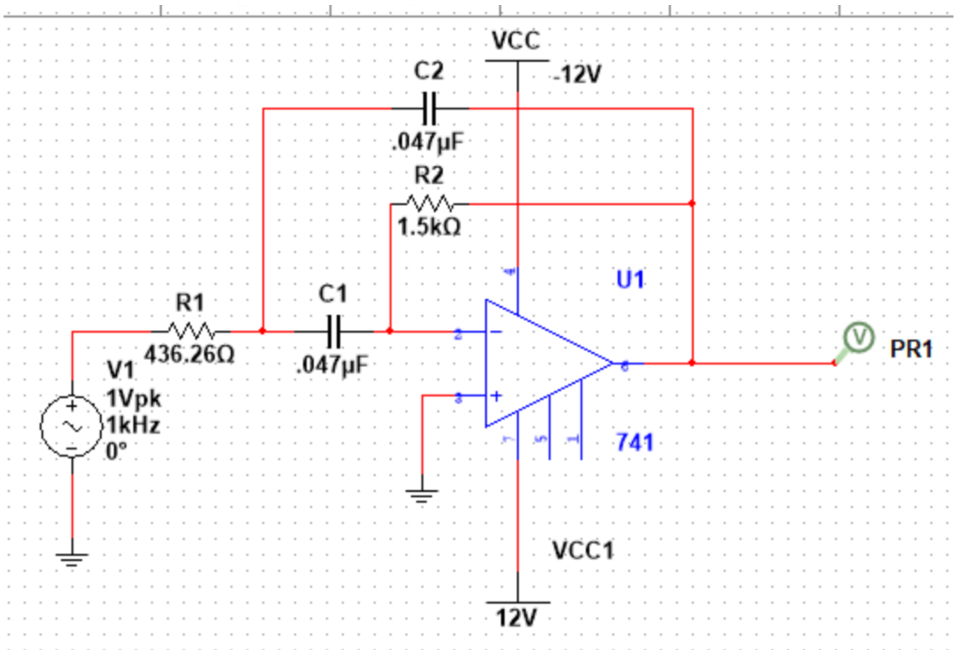
Once the circuit was validated with a spectrum analyzer, we tested it using a microphone input. The setup for this test can be seen in **Figure 12**. Using the microphone as an input source, we generated an experimental Bode plot by playing various frequencies through one of our phones. The experimental bode plot can be seen in **Figure 13**.

The results from this test again confirmed that the filter effectively isolated the C8 frequency (4186 Hz) while suppressing unwanted signals. Once we finished all the testing, we attached our filter to the audio car using adhesive tape which can be seen in **Figure 14.**

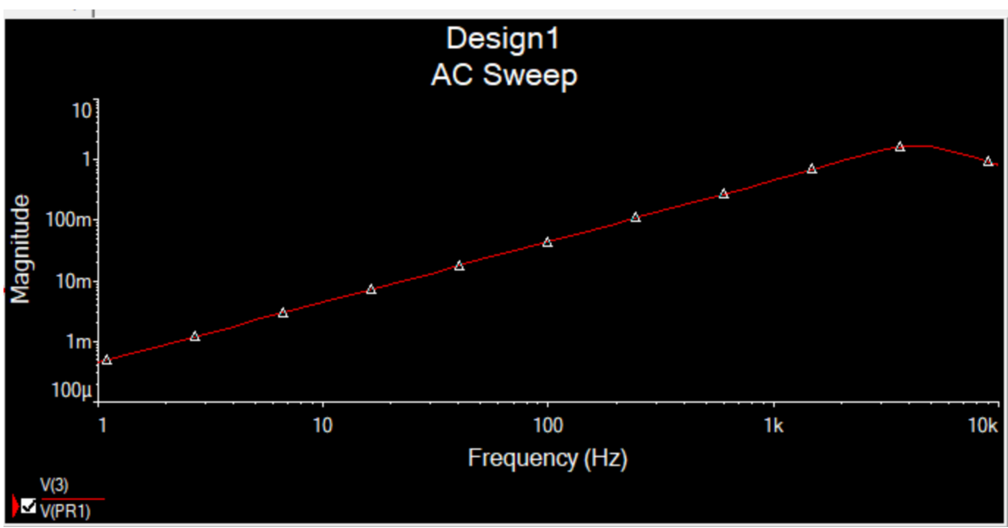
Picture(s)



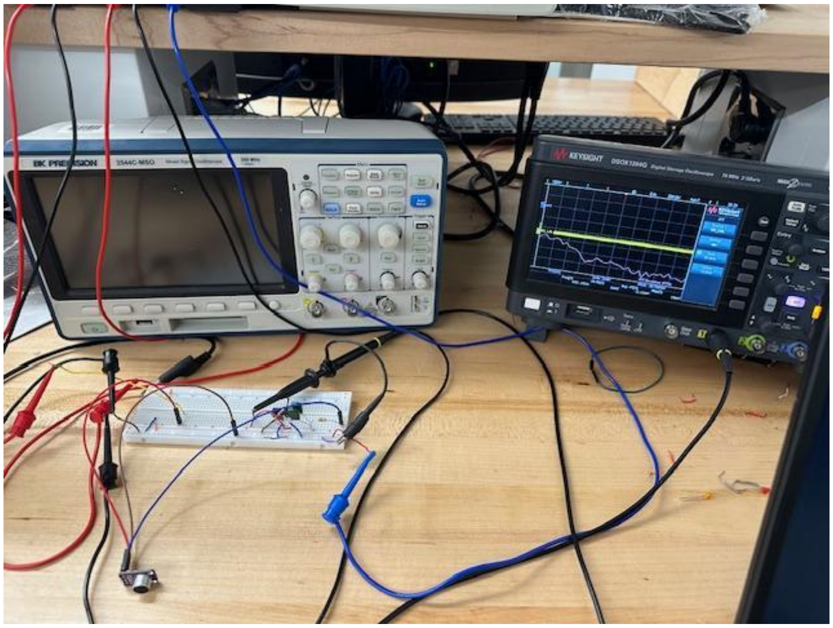
(**Figure 1:** Infinite gain Multiple Feedback Active Filter Circuit Diagram)



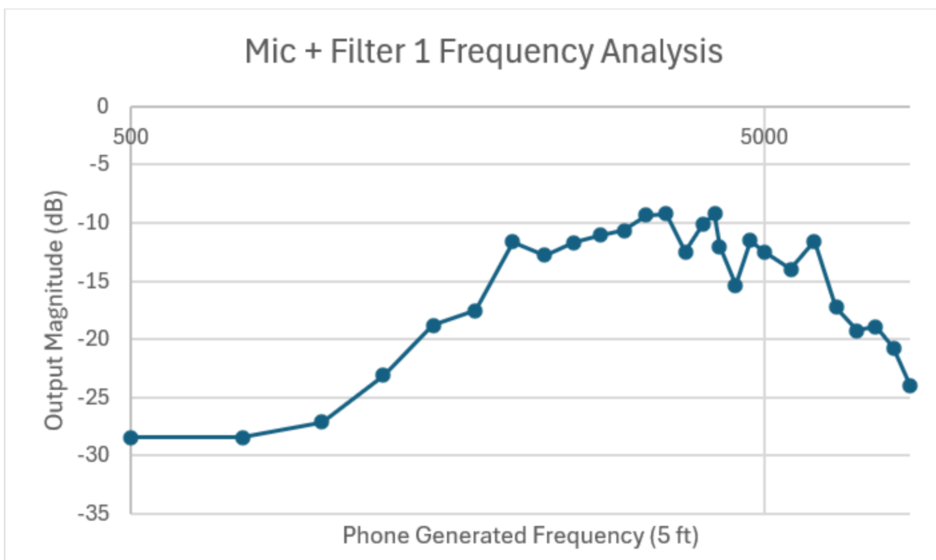
(**Figure 2:** Infinite gain Multiple Feedback Active Filter configured in Multisim.)



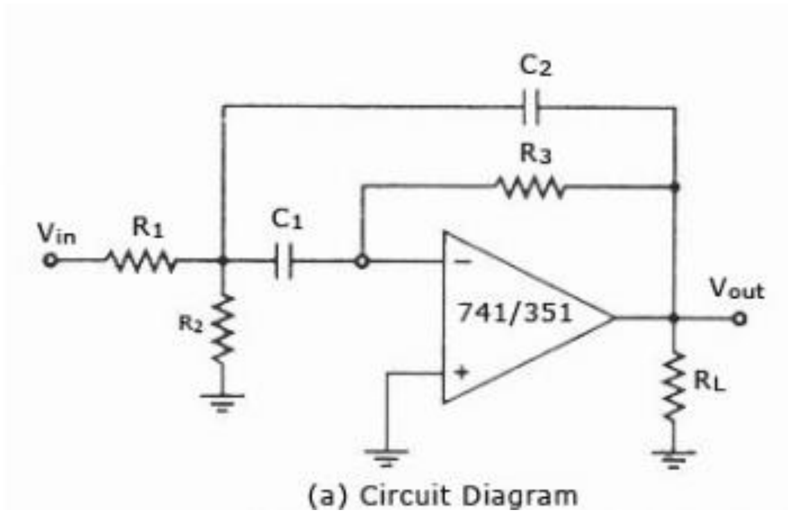
(**Figure 3:** Infinite gain Multiple Feedback Active Filter Multisim Bode Plot.)



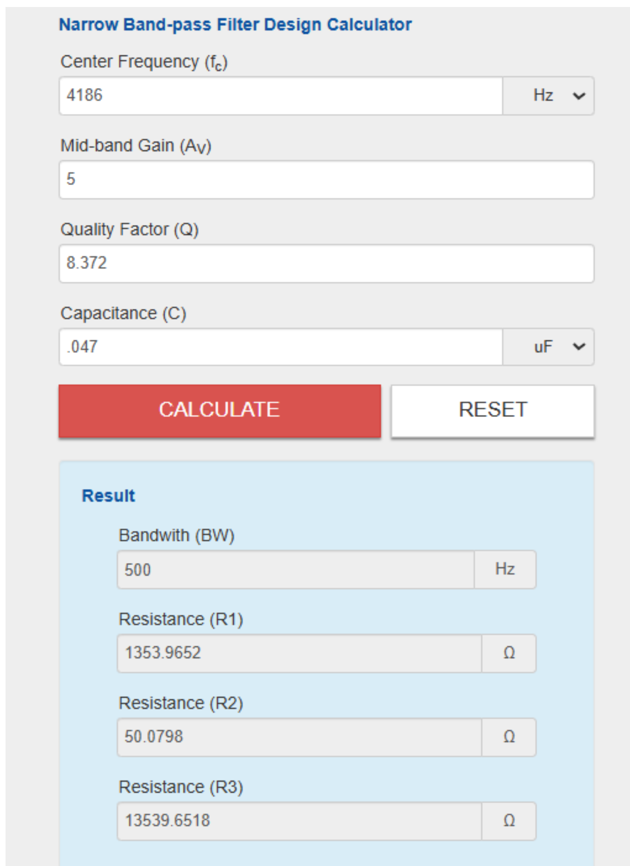
(**Figure 4:** Constructed Infinite gain Multiple Feedback Active Filter for testing.)



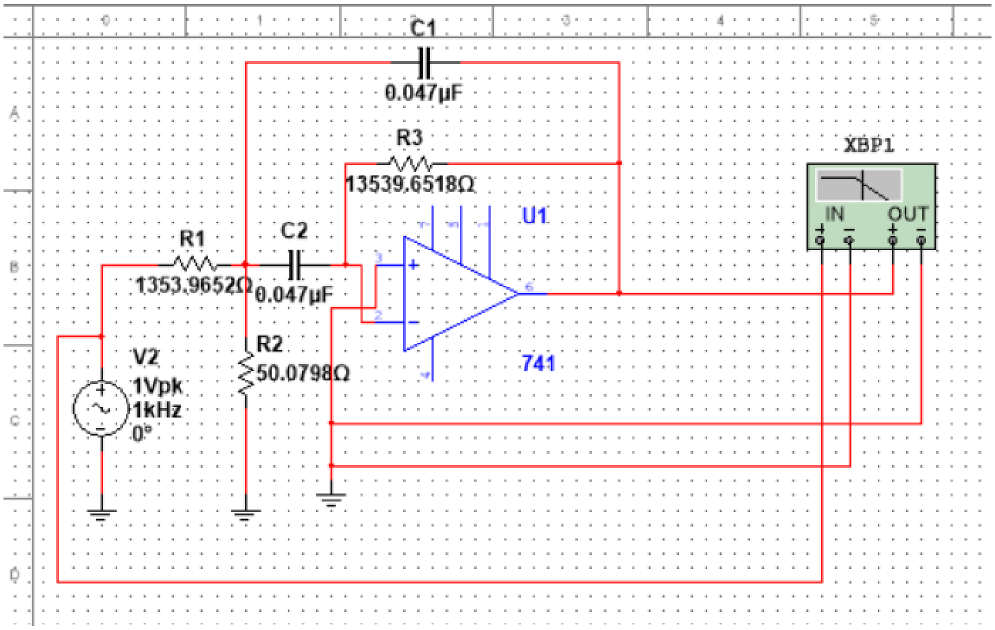
(**Figure 5:** Infinite gain Multiple Feedback Active Filter Bode plot with Microphone Input.)



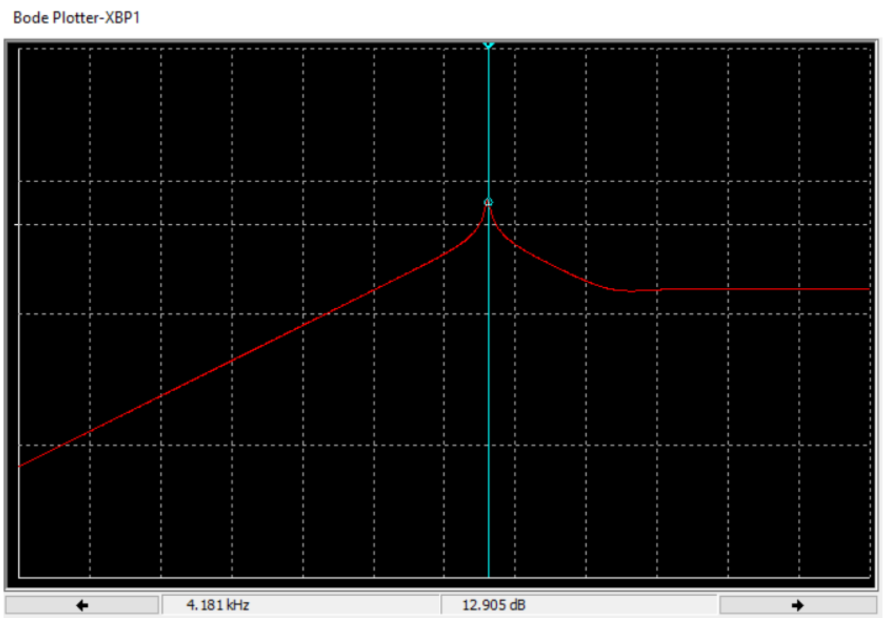
(**Figure 6:** Narrow Band-pass Filter Circuit Diagram.)



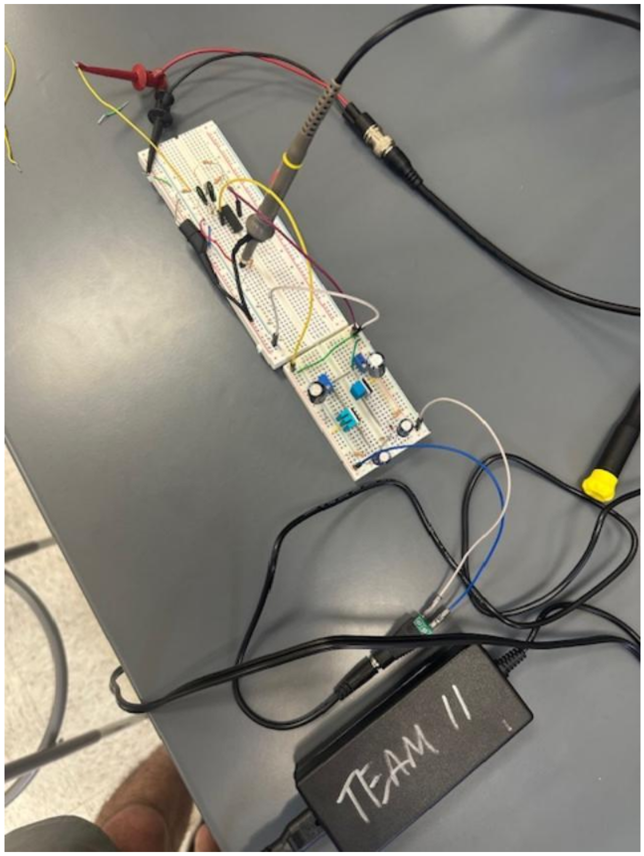
(**Figure 7:** Calculated Component Values for Narrow Band-pass Filter.)



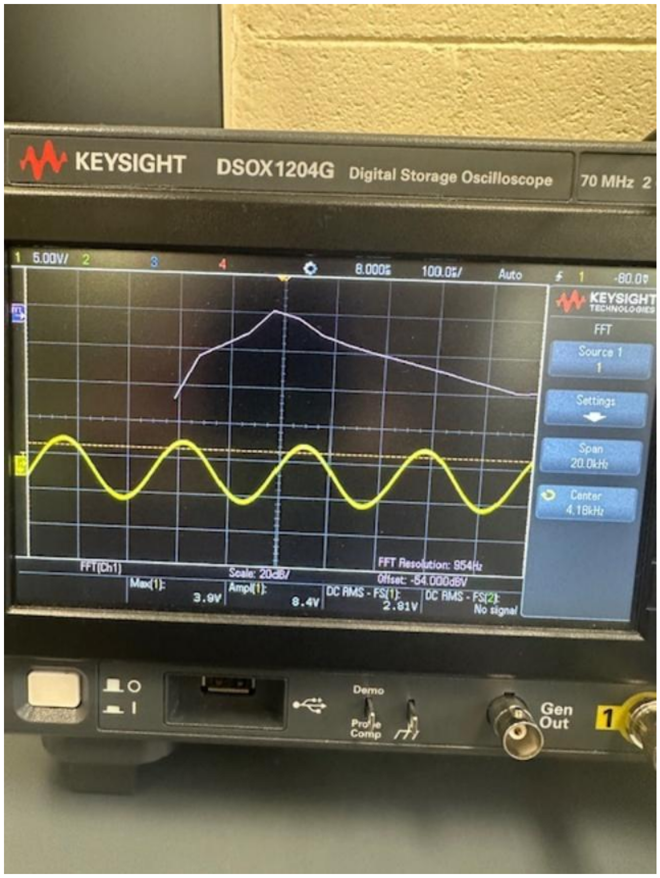
(**Figure 8:** Narrow Band-pass Filter Configured in Multisim.)



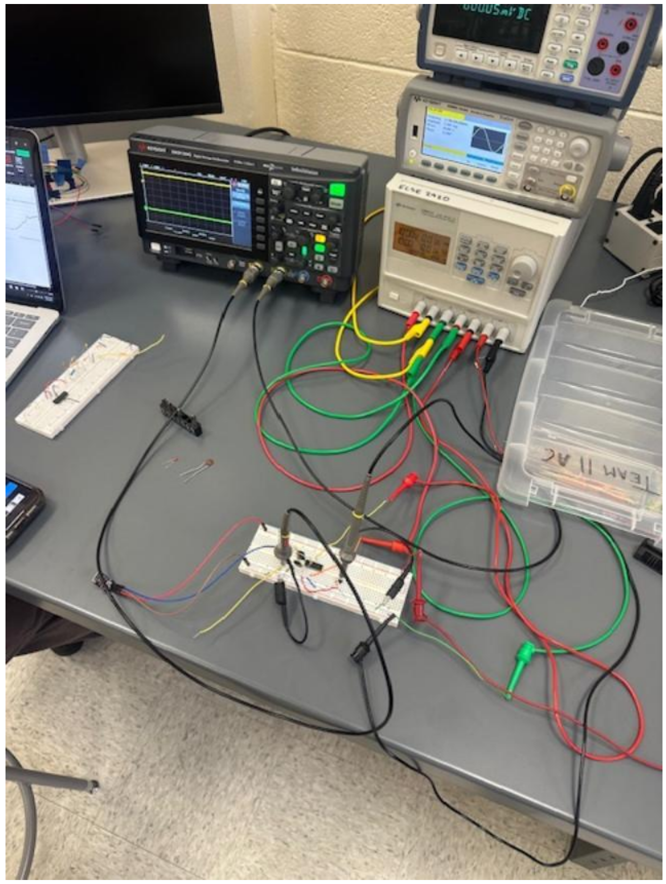
(**Figure 9:** Narrow Band-pass Filter Multisim Bode plot.)



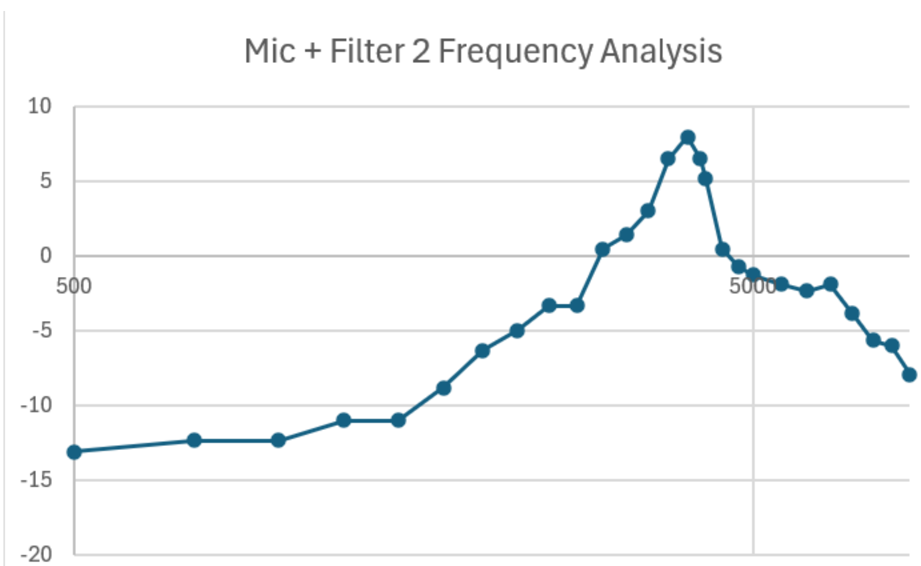
(**Figure 10:** Narrow Band-pass Filter Circuit setup with +/- 12V supply & Spectrum Analyzer Testing.)



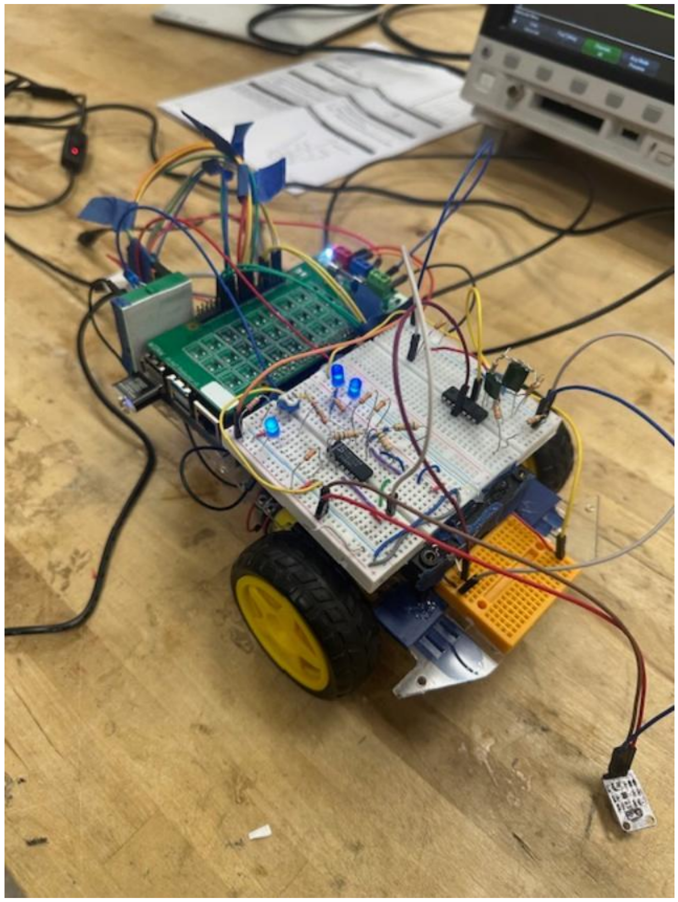
(**Figure 11:** Narrow Band-pass Filter Spectrum Analyzer Results.)



(**Figure 12:** Narrow Band-pass Filter setup for microphone input testing.)



(**Figure 13:** Narrow Band-pass Filter with Microphone Input Bode Plot.)



(**Figure 14:** Image of constructed filter on the audio car.)

Key Design Decision(s)

- What did your team clarify about the design?
  - We understood that our filter was going to need to be active and have gain to make checkpoint B easier to pass.
  - We tried to stick with just a singular filter instead of running two filters in series.
- What were the competing choices in the design?
  - Why we switched designs: Our initial Infinite Gain Multiple Feedback Active Filter did not perform well in practice. It struggled with microphone input and produced an unstable frequency response.
  - Why we chose the Narrow Band-pass Filter: More effective at isolating the target frequency, less prone to instability, component values could be precisely calculated to optimize performance, and it matched out design constraints better.

- Ultimately what did your team choose and why?
  - We chose to stick with the Narrow Band-pass filter because it produced great results through multiple tests (Multisim, Spectrum Analyzer, Microphone Input).
  - The bode plot was sharper and narrower, and overall targeted the C8 frequency better.

TEST… Test… test

- What aspects of the design need to be tested?
  - Software?
    - The software testing will come after implementing the ADC.
  - Hardware?
    - Our next step is to test the filter when the microphone and the filter itself are attached to the audio car, and where the audio is played from around 3 feet away.
- Who is responsible for testing?
  - Tests ran? Conclusions from testing?
    - Testing Summary can be seen below:

Test Scenario	Description	Outcome
Multisim Simulation (Filter 1)	Tested initial filter design in Multisim	Theoretical results were promising
Physical Circuit (Filter 1)	Built and tested first filter	Unstable performance, failed with microphone input.
Multisim Simulation (Filter 2)	Simulated the new band-pass filter	Strong theoretical response
Spectrum Analyzer Test (Filter 2)	Tested the new filter with a function generator	Successfully isolated 4186 Hz
Microphone input test	Tested with real audio input	Filtered C8 Freuqnecy effectively.

summary/part conclusion (make sure you address all parts of the requirements)

- After multiple iterations, our final design successfully meets the requirements. We implemented a Narrow Band-Pass filter using a +/- 12V power supply, verifying performance with a function generator and microphone input, measuring bode plots and comparing theoretical and experimental results. Of course, isolating the target 4186 Hz frequency.
- While our first approach failed, adapting our design to a Narrow Band-pass filter allowed us to achieve the desired performance.

Part 2: ADC Implementation and Testing

Requirements:

- **Physically build the ADC using the +/- 12V supply**
- **Apply a known analog voltage to the ADC and measure the expected output**
- **Demonstrate ADC functionality using a multimeter and known input values**
- **Provide the ADC topology and calculations showing input-output mapping**
- **Explain the resolution of the ADC and how it will be used for turning the audio car**

Attempting the R-2R ADC:

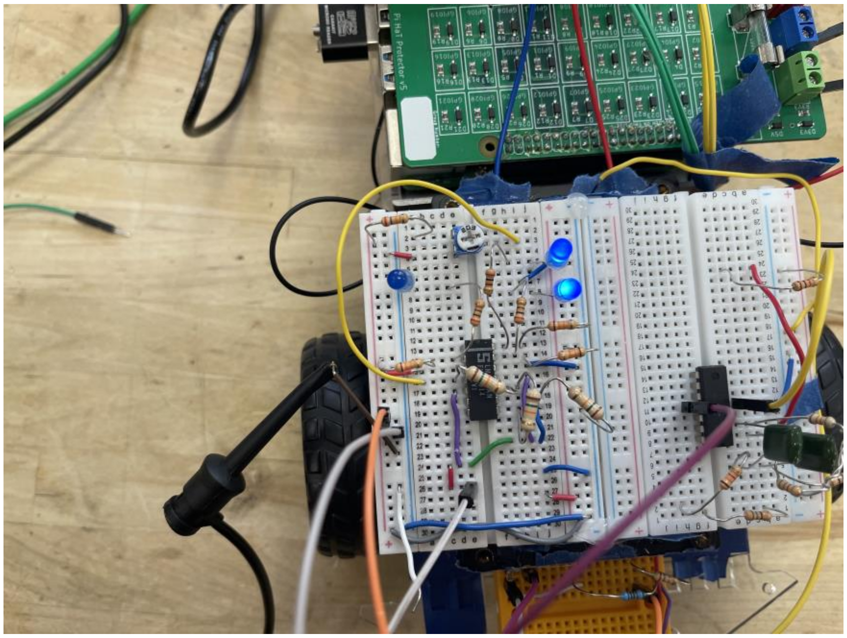
While the simulation worked ideally, the physical rendition of the R-2R ADC ended up being dysfunctional. This is because of the LM339's requirement for a pull-off resistor in the output. The input for comparator 2 on the original circuit was dependent on the output for comparator 1 and the nearby resistor values. When 10kOhm pull-off resistors were added for both comparators, the circuit didn't work as expected. We felt stuck on whether to spend time debugging this design or pivot to something new.

Redesigning and using a Resistor Ladder ADC:

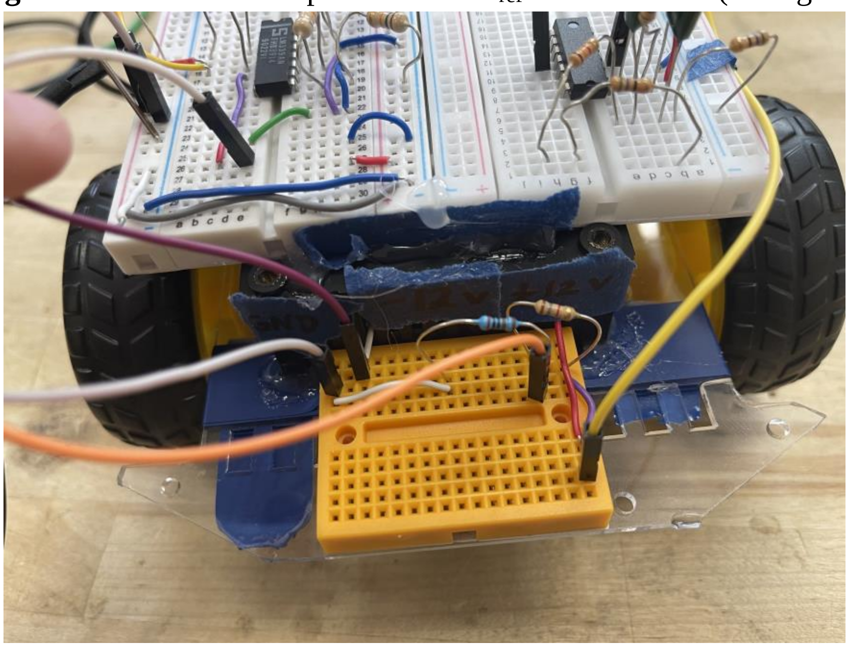
Ultimately, we decided to pivot to something new instead of possibly wasting too much time debugging the unorthodox R-2R design. We constructed a resistor ladder of 4 resistors and have each node between the ladder being compared to the reference voltage. Then each comparator output, without logic, represents a decimal count of the strength. If 2 comparators are on, then we have a strength of 2. If 1 is on, when we have a strength of 1. While this number format isn't binary, since the resolution is so small, we only need an extra GPIO to count from 0 to 3. It's also important to note that anything that needed to be grounded in the ADC will be attached to the virtual ground.

How will a 2-bit ADC be used for turning the car?

Since the Pi will only be receiving digital signals, time will be our best friend when it comes to giving context to those signals within the software side. During the checkpoint task, the car will do a full 360 to record the highest value. Once that value is found it will now turn in the same direction and record the amount of time that the strongest strength is being read for. It will then stop when the strongest strength hits a falling edge and turn in the other direction for around half of the recorded time. It will then move straight until it hits the audio source. The plan is to adjust the V_{ref} sensitivity until a "3" is hit when the microphone is facing directly at the source while the target frequency is being played. This allows us to calibrate it when the environment changes. We believe that the car will function better when the "strongest signal" time window is as small as possible.



(**Figure 15:** ADC from Top-View with V_{ref} Potentiometer (Strength 2))



(**Figure 16:** Push/Pull Power Output to ADC and Filter)

ADC Readings (Minimum Sensitivity)		ADC Readings (Medium Sensitivity)		ADC Readings (Maximum Sensitivity)	
Input Voltage	ADC Bit	Input Voltage	ADC Bit	Input Voltage	ADC Bit
0 - 0.52	0	0 - 0.26	0	0 - 0.04	0
0.53 - 1.03	1	0.26 - 0.53	1	0.05 - 0.07	1
1.04 - 1.53	2	0.54 - 0.78	2	0.07 - 0.11	2
>= 1.54	3	>= 0.79	3	>= 0.11	3

(**Figure 17:** ADC Results using Push/Pull Power and adjustable DC input)

Key Design Decision(s)

- What did your team clarify about the design?
    - **The virtual ADC failed physically. Do we redesign or debug?**
        - **If we redesign, do we implement transistor logic?**
    - **What will the reference voltage of the ADC be?**
- What were the competing choices in the design?
    - **Design Choices for ADC**
        - **New Traditional resistor ladder ADC (With Logic) (2 GPIOs)**
        - **New Traditional resistor ladder ADC (No Logic) (3 GPIOs)**
        - **Debug current ADC instead of redesigning**
    - **Design Choices for ADC Reference Voltage**
        - **Keep constant reference voltage and modify filter gain instead**
        - **Create an adjustable voltage divider for the ADC reference voltage**
- Ultimately what did your team choose and why?
    - **We ultimately chose to do a traditional ladder ADC with no transistor logic. We felt that it was easier to pick a different design than debug the previous design and how it interacts with the pull-up resistors. We also checked the amount of available GPIO ports and came to the conclusion that we could afford the extra space.**
    - **We decided to use a potentiometer to create an adjustable voltage divider for the ADC Comparators' reference voltages. We believe that this could help us readjust on the fly and modify the car's signal sensitivity when switching environments, and therefore, switching acoustic environments.**

TEST… Test… test

- What aspects of the design need to be tested?
    - Software?
        - **We need to test if the ADC can create digital signals to the pi**
    - Hardware?
        - **We need to test the ADC in conjunction with the push/pull, filter, and microphone**
            - **Also needs to be tested on the car as well.**
- Who is responsible for testing?
    - Tests ran?
        - **The ADC was powered with the push/pull and used a DC input as the input signal. It was then mapped at three sensitivity levels and was matched to the output voltage bode plot from our filter.**
        - **We measured the output voltage of the ADC in respect to the raspberry pi ground.**
    - Conclusions from testing?
        - **The ADC works as intended and matches the desired max strength voltage value of ~1.5V that the filter is expected to output.**

    - **The measured voltage was far above 3V (Which implies the virtual ground is at play) and therefore cannot be currently plugged into the Pi.**

Summary / Part Conclusion:

- **The original R-2R ADC design failed due to issues with the LM339 comparator requiring pull-up resistors. Instead of debugging, we opted for a traditional resistor ladder ADC without transistor logic, utilizing three GPIOs for simplicity and reliability.**
- **A potentiometer-based adjustable voltage divider was chosen for the ADC reference voltage, allowing on-the-fly calibration in different environments. The ADC was tested with a push/pull power setup and a DC input, confirming that it produces expected digital outputs.**
- **While the ADC successfully maps signal strength, the measured voltage is too high for direct Raspberry Pi input. This is next on our to-do list.**

While we got everything to work in tandem on its own, I believe that there could be trouble implementing the full system to the Pi without damaging it. We must proceed with caution and put our thoughts behind every action made.

Conclusion and Participation (REQUIRED)

1. **Include a selfie/photo from your group meeting/zoom call.**

2. **When did you meet?**
   3/10, 3/11, and 3/12
3. **Who was present?**
   Dawson Gulasa, Finn Morton, Michael See, Cade Toney
4. **Who was not present?**
   Every member was present
5. **What were the main ideas discussed or major decisions (1-2 sentences/bullet points)**
   a. What to do about the ADC overhaul
   b. How to adjust the car’s sensitivity/reference voltage